DroMOOC drone simulator

The *simulator* ROS package implements a dynamic model of a quadrotor, position and attitude controllers as well as graphical tools for parameter tuning and visualization.

Detailed instructions on how to install and run the Virtual Machine can be found in the document "Importing and starting the DroMOOC Virtual Machine".

The *simulator* package is located in the `~/catkin_ws/src/simulator` directory of the Virtual Machine and in the DroMOOC Github repository. Before first use, it is recommended to update the package located on your Virtual Machine by entering the following command lines in a terminal (internet connection is required):

```
roscd simulator
git pull
```

If you need some help, use the contact us link on the website.

1. Start the simulator

Open a new terminal, and enter the following command line:

```
roslaunch simulator quadrotorSimu.launch
```

This command line will start a ROS master, the simulation node (`quadrotorSimu.py`), the position and the attitude controller nodes (`positionCtrl.py` and `attitudeCtrl.py`).

This command line also starts a graphical user interface based on RQT (Figure 1), which enables to set references values for the position (WayPoint) and yaw angle (YawRef) of the drone (Figure1 – top-left), to tune parameters of the position and attitude controllers (Figure 1 – top-right), and to plot the time evolution of attitude angles (Figure 1 – bottom-left) and position coordinates (Figure 1 – bottom-right) of the drone.

A 3D visualization (Figure 2) presents the drone and its own reference frame, the global fixed reference frame and the current Way Point defined for the drone (yellow marker).

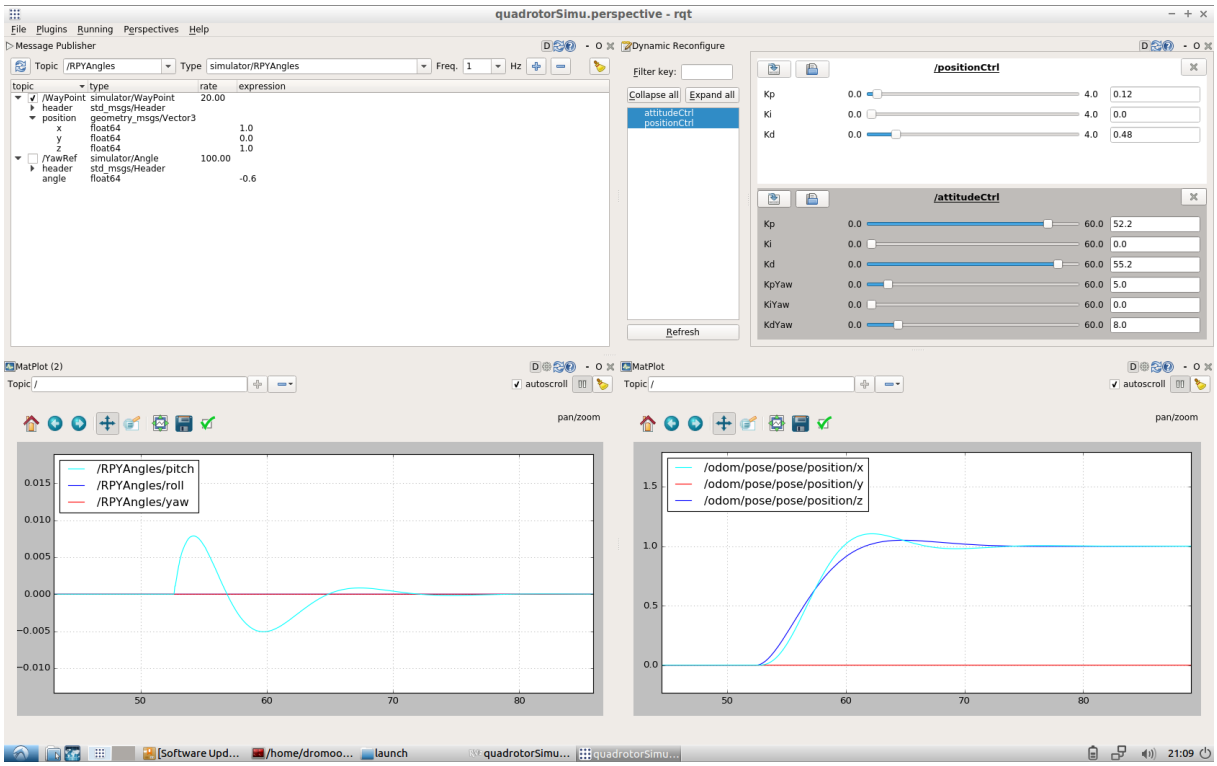


Figure 1 – Graphical user interface of the simulator

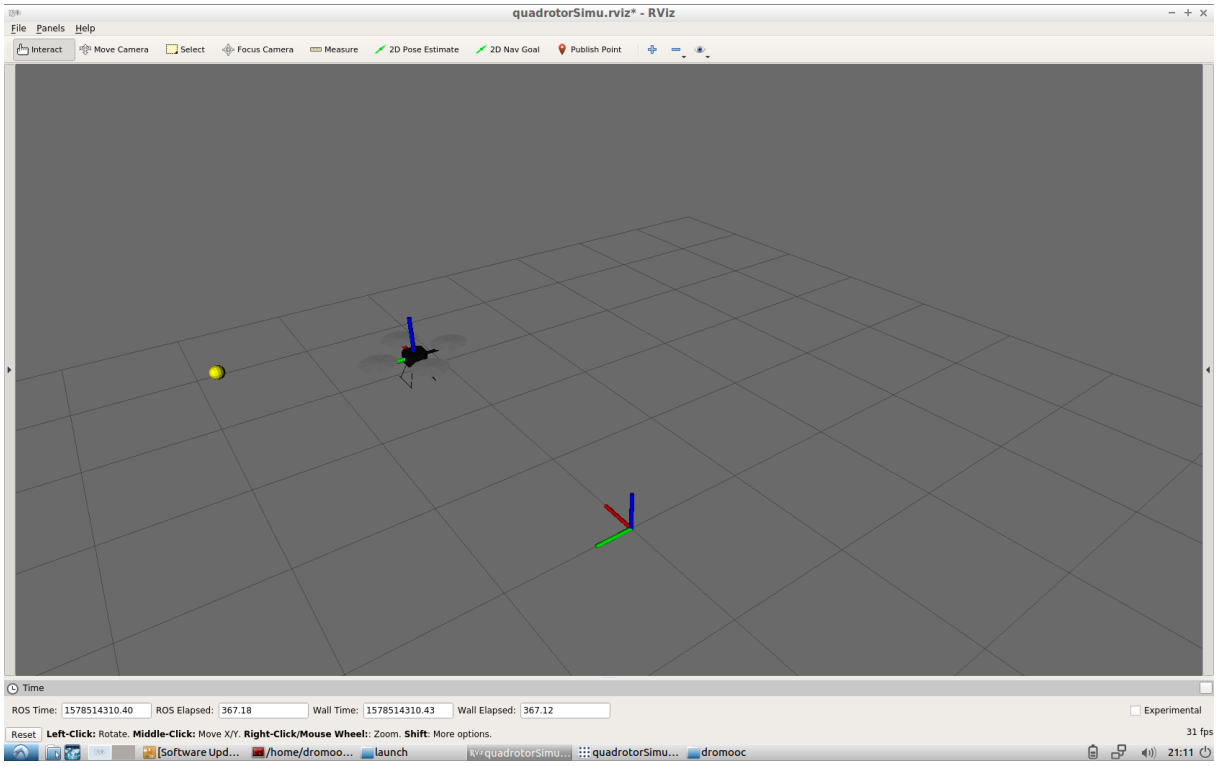
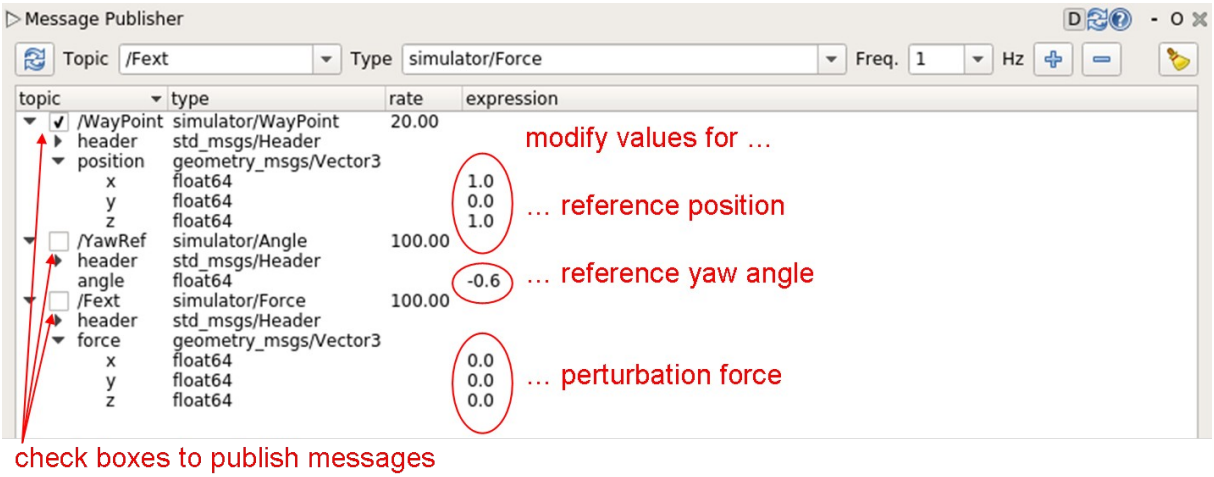


Figure 2 – 3D visualization

2. Set reference values and tuning parameters

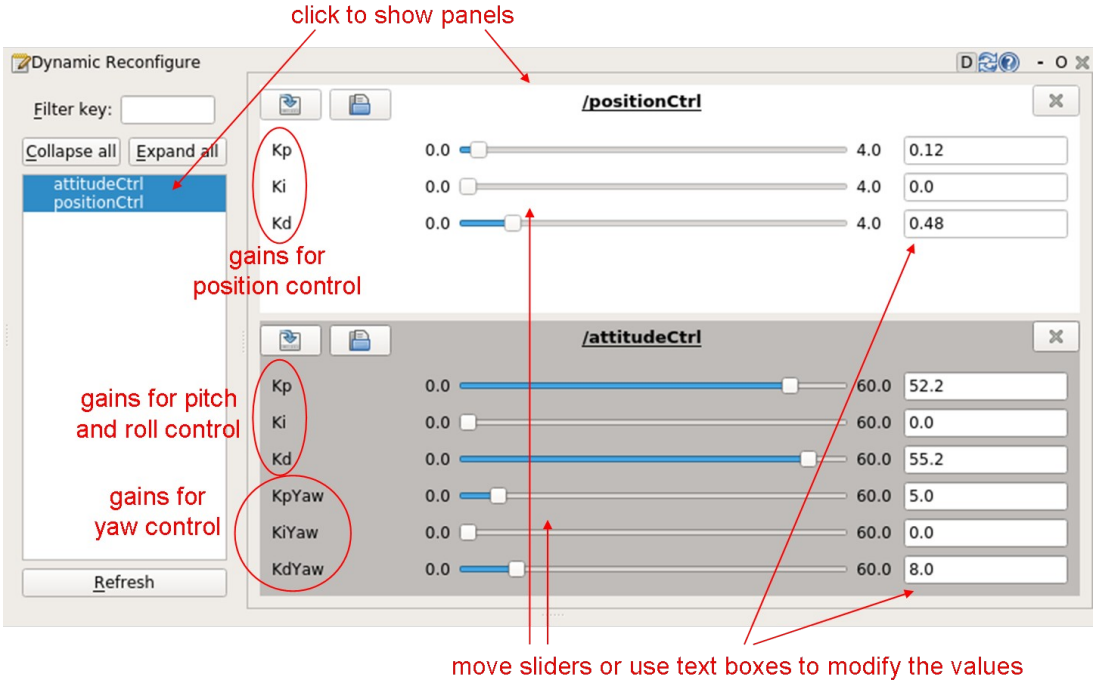
The Message Publisher interface can be used to set the reference values for the position and the yaw angle.



Enter the desired values in the fields position/x, y,z of the /WayPoint topic to define the reference position for the drone. Check the box corresponding to the /WayPoint topic to make it published to the position controller.

Enter the desired value in the field angle of the /YawRef topic to define the reference position for the drone. Check the box corresponding to the /YawRef topic to make it published to the controllers.

The Dynamic Reconfigure interface can be used to tune the gains of the position and attitude controllers while the simulation is running.



Note that PD controllers are implemented, therefore modifications of the integral gain will have no influence unless an integral action is implemented in the code (see Sections 3 and 4 of this document).

3. Understanding the code

The simulator package is composed of several directories:

- `scripts`: Python files for simulator node, position control node and attitude control node
- `msg`: custom message formats used by the simulator and controllers
- `launch`: launch files that are used to run several nodes
- `doc`: this document

Proportional Derivatives (PD) controllers are implemented for position and attitude controllers. Development of an integral action is left as "homework" (see Section 4 of this document).

The position controller requires position and velocity measurements (/odom topic), reference position (/WayPoint topic), reference yaw angle (/YawRef topic) and computes the magnitude of the thrust (/thrust topic) and references angles for the attitude controller (/RPYAnglesRef topic).

The attitude controller requires angle and angular velocity measurements (/odom topic), reference angles (/RPYAnglesRef topic) and computes the torque (/torque topic).

The thrust and torque computed by the controllers are applied to the simulation model. A perturbation force (/Fext topic) can also be applied to the drone.

The main structure of the simulator is summarized by the following Node Graph automatically generated by ROS:

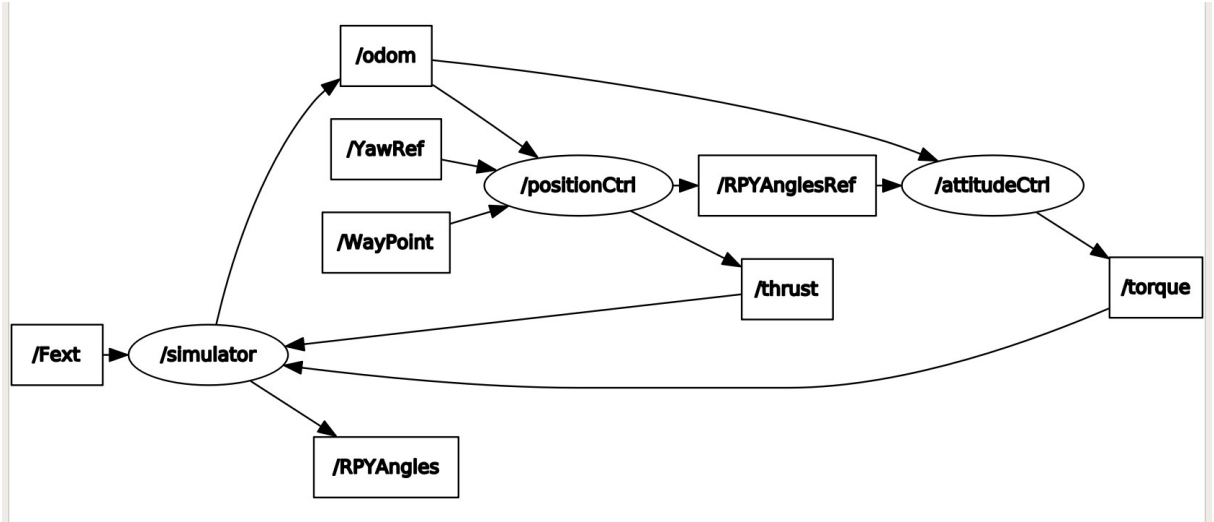


Figure 3 – Node Graph of the simulator and controllers

4. Proposed work (“homework”)

The simulator package is provided as standalone additional material to DroMOOC to understand the basis of drone control and simulation. Two simple and direct improvements are proposed below as “homework”.

Integral action

Modify the position controller node (positionCtrl.py) to add an integral action in the control law. Test the effect of the integral action by applying a constant perturbation force (/Fext topic) to the drone during the simulation.

Way point navigation

Develop a new node that publishes a reference position to the drone (/WayPoint topic), given a list of pre-defined Way Points and information on the current position of the drone (/odom topic).

If you need some help, use the contact us link on the website.